BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 20 2005, pages 3824–3831
doi:10.1093/bioinformatics/bti627

Sequence analysis

Statistical evaluation of pairwise protein sequence comparison with the Bayesian bootstrap

Gavin A. Price[1], Gavin E. Crooks[2], Richard E. Green[2] and Steven E. Brenner[1,2,*]

[1]Department of Bioengineering and [2]Department of Plant and Microbial Biology, University of California, Berkeley, CA 94720, USA

Received on April 14, 2005; revised on July 16, 2005; accepted on August 11, 2005
Advance Access publication August 16, 2005

ABSTRACT

Motivation: Protein sequence comparison methods are routinely used to infer the intricate network of evolutionary relationships found within the rapidly growing library of protein sequences, and thereby to predict the structure and function of uncharacterized proteins. In the present study, we detail an improved statistical benchmark of pairwise protein sequence comparison algorithms. We use bootstrap resampling techniques to determine standard statistical errors and to estimate the confidence of our conclusions. We show that the underlying structure within benchmark databases causes Efron's standard, non-parametric bootstrap to be biased. Consequently, the standard bootstrap underpredicts average performance when used in the context of evaluating sequence comparison methods. We have developed, as an alternative, an unbiased statistical evaluation based on the Bayesian bootstrap, a resampling method operationally similar to the standard bootstrap.

Results: We apply our analysis to the comparative study of amino acid substitution matrix families and find that using modern matrices results in a small, but statistically significant improvement in remote homology detection compared with the classic PAM and BLOSUM matrices.

Availability: The sequence sets and code for performing these analyses are available from http://compbio.berkeley.edu/.

Contact: brenner@compbio.berkeley.edu

INTRODUCTION

The workhorse method of computational protein sequence analysis is pairwise alignment (Needleman and Wunsch, 1970; Smith and Waterman, 1981; Gotoh, 1982; Durbin *et al*., 1998). This is the underlying methodology of programs such as SSEARCH (Pearson, 1991), FASTA (Pearson *et al*., 1988) and BLAST (Altschul *et al*., 1990). In order to compare, analyze, parameterize and improve both existing and novel sequence algorithms, it is first necessary to accurately measure their effectiveness (Henikoff and Heinkoff, 1993; Brenner *et al*., 1995; Murzin *et al*., 1995; Pearson, 1995; Gribskov and Robinson, 1996; Pearson, 1996, 1998; Brenner *et al*., 1998; Geetha *et al*., 1999; Schaffer *et al*., 1999; Blake and Cohen, 2001; Schaffer *et al*., 2001; Green and Brenner, 2002). To this end, we had previously developed a sensitive and flexible system for the evaluation of pairwise protein sequence comparison (Brenner *et al*., 1998; Green and Brenner, 2002). We first assembled a non-redundant dataset of several thousand protein sequences from the SCOP (structural classification of proteins) database (Murzin *et al*., 1995) whose evolutionary relations have been reliably determined using structure, function and sequence. Every sequence in the dataset is then aligned against every other sequence using the algorithm under investigation. The resulting alignment scores (typically *E*-values) are used to rank order the matches. A threshold is then drawn. There is a natural tradeoff between allowing few false

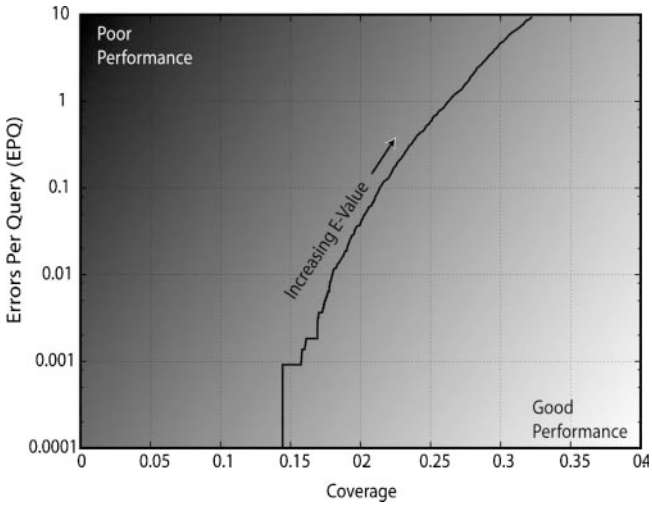

Fig. 1. Example coverage versus errors per query (CVE) plot. The coverage is the fraction of homologous sequences detected at the corresponding error rate, which is the average number of false positives for each database query. As we become more permissive with respect to what constitutes a sequence match (increasing the cutoff of the program's reported *E*-Value), the program finds more matches at the expense of producing more errors. Thus the coverage and error per query (EPQ) both increase. CVE plots may be thought of as ROC plots with the axes swapped and scaled suitably to focus attention on the most relevant error values.

*To whom correspondence should be addressed at Department of Plant and Microbial Biology, 111 Koshland Hall #3102, University of California, Berkeley, CA 94720-3102, USA

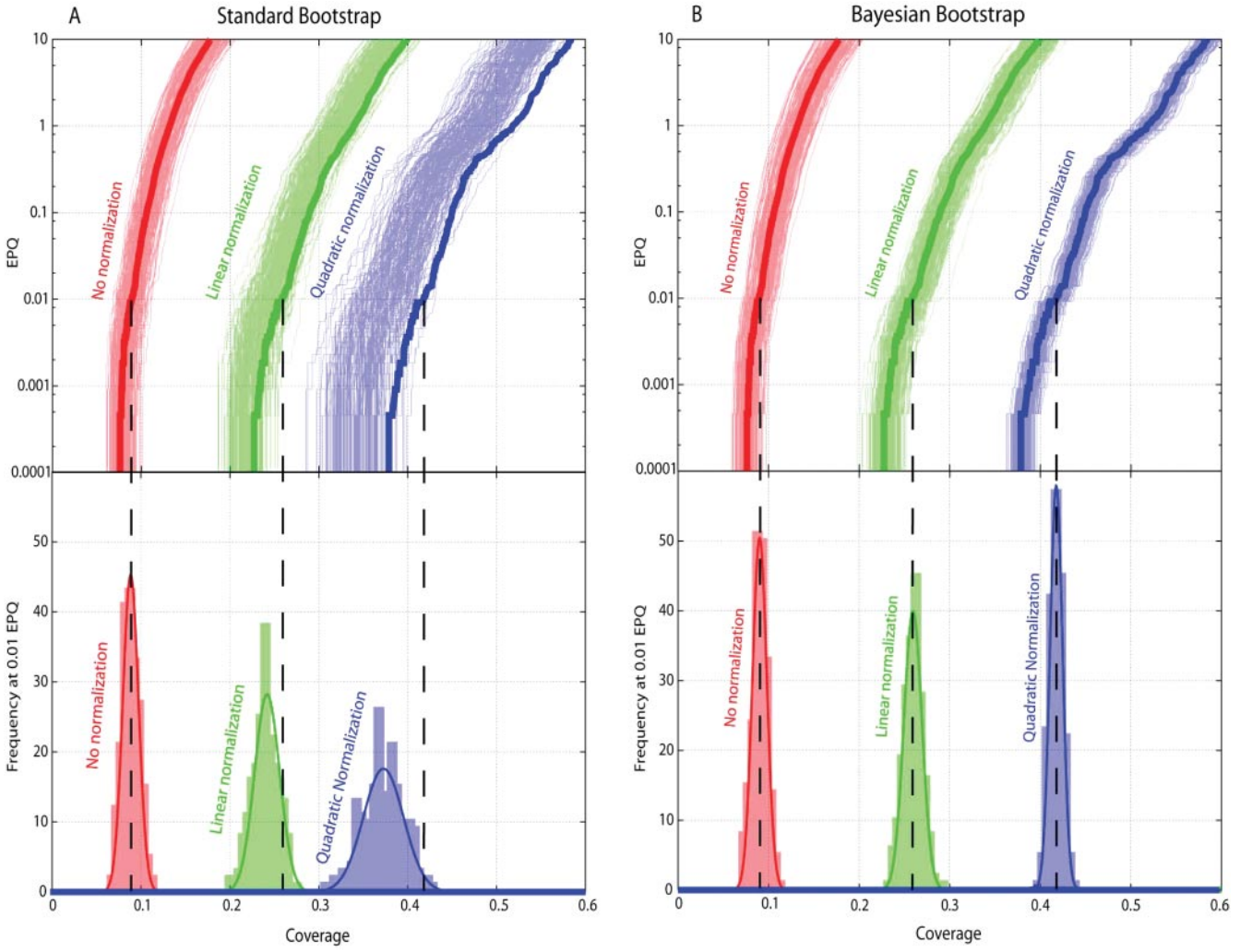

Fig. 2. Demonstration of the difference between a standard and Bayesian bootstrap using the optimal parameter settings for the BLOCKS 13+ BLOSUM matrix family and the test dataset. The original CVE lines are heavier and thicker than the 200 bootstrapped replicas. (**A**) The standard bootstrap preferentially selects sequences from larger, more diverse superfamilies where the correct sequence relationships are harder to discover. Thus, when each superfamily possesses the same amount of possible coverage (quadratic normalization), the bootstrap is biased toward the left because smaller superfamilies often drop out of the analysis entirely. Linear normalization displays a less severe effect. Since larger superfamily relationships are harder to discover, when the superfamilies have equal total weight (quadratic normalization), the coverage is much higher than with no normalization. To a lesser degree, the same effect is observable with linear normalization. The bottom graph makes clear that the standard bootstrap also overpredicts the variance under normalization. (**B**) As the Bayesian bootstrap assigns non-integer weights to each sequence, smaller superfamilies will not drop out of the analysis. This eliminates the bias and overpredicted variance of the standard bootstrap.

positive errors with a stringent score cutoff, or allowing more errors but also finding more true relations with a more permissive threshold. We, therefore, vary the score threshold, and plot the proportion of true relations found (homology coverage) versus the number of false positive matches (errors per query or EPQ) in a manner conceptually similar to ROC plots. This is illustrated in Fig. 1. However, this unnormalized coverage is dominated by the largest SCOP superfamilies, since the number of relations scales as the square of the superfamily size. To compensate for this unwarranted dependence, we also report reweighted results, namely the average fraction of true relations per sequence (linear normalization) and the average fraction of true relations per superfamily (quadratic normalization) (Green *et al*., 2002).

However, it is not sufficient to only determine the difference in performance of two algorithms. It is also necessary to determine if the observed differences are statistically significant, given the finite size of our datasets. To this end, we previously estimated standard statistical errors and confidence intervals using the non-parametric bootstrap resampling method of Efron (Efron 1979; Efron and Robert, 1993). We generate many replicas of the original dataset by sampling N sequences, with replacement, from our original dataset of N sequences. We then calculate the statistic of interest (typically, the homology coverage at 0.01 EPQ) for each replica. The standard deviation of the replica statistics is an approximation to the standard error induced by the finite size of our dataset.

Unfortunately, this straightforward resampling technique leads to anomalous results when applied to our evaluation of pairwise protein sequence comparison, illustrated in Figure 2A. As can be seen, the coverage versus error (CVE) lines of the replicas (thin) are biased relative to the original data (thick), underestimating the true coverage. Why should this be? On reflection, it is apparent that these anomalies are the result of an unfortunate interaction between the resampling procedure and the fine structure of our dataset. Within SCOP, related protein domains are grouped into superfamilies. There are a few large, and many very small superfamilies (Green and Brenner, 2002). As a result of the sampling procedure used in the standard bootstrap, each sequence is represented zero, one or more times. Because the dataset is moderately large, sequence weights are approximately Poissionian with unit mean. Therefore, the chance of not including a particular sequence in a replica is $\sim$1/e, or 37%. The chance of including both sequences of a size-2 superfamily at least once is 40%. In other words, since self-relations are not considered in our analysis, each replica has a 60% chance of entirely neglecting each size-2 superfamily. Smaller, but still significant proportions of other small superfamilies are also dropped. Large superfamilies, however, may change in size from sample to sample, but there is a proportionally smaller chance that they will fail to be represented. This fact alone would not introduce a bias into the standard bootstrap procedure if the relationships within smaller superfamilies were, on average, just as easily detectable as those within larger superfamilies. However, this is not the case. In general, proteins within small superfamilies are more closely related, and less diverse, than those within large superfamilies, and easier to detect using pairwise sequence comparison. Consequently, the undersampling of small superfamilies that results from using the standard bootstrap biases the samples toward larger, more difficult superfamilies, leading to an unwarranted reduction in homology coverage within the replica ensemble.

One possible solution to this biased sampling of superfamilies might be to resample entire superfamilies, rather than individual sequences. However, this will probably lead to very noisy replica ensembles, since the largest superfamilies contain the majority of the intersequence relations. Another potential problem is that resampling on superfamilies assumes that relations are transitive (i.e. if A is related to B, and B is related to C, then A and C are related) and therefore, related sequences are grouped into non-overlapping families. This property is true for our current dataset, since SCOP separates proteins into evolutionary domains. But many interesting datasets are not transitive (e.g. multidomain protein sequences), and therefore superfamily resampling is not universally applicable.

Schaffer *et al*. (2001) presented a related bootstrap method for the evaluation of PSI-BLAST performance. Instead of resampling all sequences, they resample only the false positives. This allows an analytic evaluation of the bootstrap distribution, obviating the computational costs, but this approximation may introduce unnecessary bias to the error calculations.

As an alternative, we have implemented the Bayesian bootstrap (Rubin, 1981), a Bayesian resampling procedure that is operationally similar to the standard non-parametric bootstrap. In the standard bootstrap, resampling with replacement in effect assigns to each sequence integer weights drawn from a multinomial distribution. In the Bayesian bootstrap, the sequences are assigned continuously varying weights drawn from a Dirichlet distribution. This alternative procedure has a clear Bayesian interpretation. In essence, we assume that the sequences have been sampled from some unknown distribution to which, in the absence of any pertinent information, we associate an uninformative prior. This prior combines with the multinomial sample likelihood, via Bayes' theorem, to result in a Dirichlet posterior distribution on the fraction of the original population that each sampled sequence represents. Therefore, we can think of the ensemble of Bayesian bootstrap replicas, and the distribution of statistics derived from them, as samples from a Bayesian posterior distribution (Durbin *et al*., 1998).

In practice, we find that the Bayesian bootstap does not suffer from the strong replica bias exhibited by the standard bootstrap (Fig. 2B). Why should this be? The standard bootstrap has a 37% chance of not including any given sequence in a replica. Consequently, its resampling does not preserve the structure of relations between and within superfamilies, which has a detrimental effect on the CVE statistic, since it is sensitive to this structure. In contrast, in the Bayesian approach the sample weights are continuously varying, and therefore there is a vanishingly small chance of assigning a zero weight to any sequence. Thus, all of the interrelations between sequences are preserved in the replicas, albeit reweighted (in particular, we no longer undersample small superfamilies), and the replicas provide a more trustworthy estimate of the inherent uncertainly in our statistic due to the finite size of the dataset.

We have previously used our sequence comparison evaluation to contrast various alignment programs, including BLAST, FASTA and SSEARCH, to select appropriate gap parameters, and to rigorously evaluate statistical *E*-value homology scores (Brenner, 1996; Brenner *et al*., 1998; Park *et al*., 1998; Green and Brenner, 2002). More recently, we have applied the Bayesian bootstrap statistics described herein to compare different protein sequence gapping models (Zachariah *et al*., 2004), to contrast various models of amino acid evolution (Crooks and Brenner, 2004) and to evaluate algorithmic extensions to standard Smith–Waterman alignment (Crooks, Green and Brenner, 2005).

As a concrete example of using our methodology, in this paper we compare the performance of several different substitution matrix families. Every pairwise sequence alignment program requires a substitution matrix, a 20×20 table of scores, each of which represents the propensity for some amino acid to be replaced by a different amino acid during the course of protein sequence evolution. A matrix family encompasses a set of matrices that are suitable for different evolutionary distances. In principle, we should match the divergence inherent in the substitution matrix to the divergence of the pair of sequences we wish to align (Altschul, 1993). However, this is computationally expensive, and, in practice, a single matrix is chosen based on its ability to align remote homologs, on the grounds that matching close homologs is relatively easy (Brenner, 1996).

Many different matrix families have been created using different datasets and different evolutionary models. In the present work, we compare the relative effectiveness of four such families. The popular BLOSUM matrices were derived empirically from the BLOCKS database of reliable protein sequence alignments (Henikoff and Henikoff, 1992; Henikoff *et al*., 2000). We have also created a family of BLOSUM matrices reparameterized using the BLOCKS 13+ database. This BLOCKS version contains many more sequences than BLOCKS 5, the database version used to create standard BLOSUM matrices.

The classic PAM matrices (Dayhoff, 1978) were trained on a limited set of close homologs using a Markovian model of amino acid replacement. The modern VTML (variable time maximum likelihood) matrices are also based on the Dayhoff model, but are trained on a large set of diverse homologs (Muller and Vingron, 2000; Muller *et al*., 2002).

A major advantage of the PAM and VTML families is that since they are directly associated with an explicit model of amino acid substitution, they can be used for distance estimation and maximum-likelihood tree estimation procedures. In contrast, there is no unique rate matrix associated to the BLOSUM matrixes. After extensively testing these four families, we find that the modern VTML and BLOCKS 13+ BLOSUM matrices show a small, but statistically significant improvement in remote homology detection compared with the classic PAM and BLOSUM matrices, respectively.

MATERIALS AND METHODS

Dataset construction

A set of proteins whose evolutionary interrelations are known was assembled from the SCOP database (version 1.61) (Murzin *et al*., 1995). SCOP classifies protein domains using structure, function and sequence and has been widely used as a gold standard for interprotein evolutionary relations (Brenner *et al*., 1995; Russell *et al*., 1997; Brenner *et al*., 1998; Karplus *et al*., 1998; Lindahl and Elofsson, 2000). Protein structures are divided into separate evolutionary domains, which are then classified into a hierarchy of class, fold, superfamily and family. Homologous domains are placed into the same superfamily, whereas domains belonging to different classes or folds may safely be considered unrelated. We treat the evolutionary relationship of domains classified in the same fold but different superfamilies as undetermined and do not consider them in our benchmarking (Green and Brenner, 2002). To focus our evaluations on the detection of remote homologs, rather than highly similar sequences, we filter the protein domains such that no two sequences share >40% pairwise identity. The ASTRAL compendium (Brenner *et al*., 2000; Chandonia *et al*., 2002, 2004) conveniently provides such SCOP subsets. The 40% filtered set was further divided into training (2592 sequences) and test (2182 sequences) sets—this allows for optimization of methods on the training database and comparison on the test database to avoid overfitting. The training set consists of the odd numbered folds in SCOP classes a, c, e and g, and even folds from classes b, d and f, and conversely for the test set. This alternation of folds is necessary to obtain approximately equally sized subsets, since the first fold in a class is generally the largest. Use of distinct folds maintains the independence of the sets.

Superfamily size normalization

The number of relationships within a given superfamily grows quadratically with the size of the superfamily. Therefore, the large superfamilies account for most true relations between protein sequences in the database analysis. This is potentially problematic, since there are known biases within the database of solved protein structures (and by extension, within SCOP and ASTRAL) and between superfamilies. In particular, the protein domains within large superfamilies are more diverse, and the interrelations harder to discover, than proteins within small superfamilies. Because of this bias, and the dominance of large superfamilies, performance evaluations may be skewed.

In order to compensate for these effects, we previously developed two alternative normalization methods (Green and Brenner, 2002). In linear normalization the weight of each sequence match is divided by the number of true homologs of the query (i.e. $s-1$, where s is the size of the superfamily). In quadratic normalization the weight of each sequence match is divided by the total coverage of the superfamily, i.e. the number of true relations within the superfamily, (s^2-s). In other words, unnormalized coverage is the fraction of all true relations that are found, linear normalized coverage is the average fraction of true relations per sequence and quadratic is the average fraction per superfamily. Since linear and quadratic normalizations systematically downweight large superfamilies relative to small superfamilies, and because finding correct relations in large superfamilies is harder, quadratic coverage is generally larger than linear coverage, which in turn is larger than unnormalized coverage, as can be seen in Fig. 2.

Non-parametric and Bayesian bootstrap

The statistical errors and statistical significance of homology coverage were estimated using both non-parametric and Bayesian bootstrap resampling. In Efron's standard, non-parametric bootstrap (Efron, 1979; Efron *et al*., 1993), replicas of the original dataset are generated by sampling N items, with replacement, from the original dataset of size N. Equivalently, in each replica the items are assigned integer weights, 0, 1, 2 . . . , distributed according to the multinomial distribution. The distribution of the statistic of interest across the ensemble of replicas is taken as an estimate of the statistical errors owing to the finite size of the original dataset.

In the alternative Bayesian bootstrap (Rubin, 1981) the data items in each replica are assigned continuous weights drawn from a Dirichlet distribution. This Dirichlet can be thought of as the posterior distribution of sequences in the original population, assuming multinomial sampling of the original data from an improper prior across all possible sequences (i.e. *a priori* all sequences are equally probable, but since there are many possible sequences, each has vanishing small weight in the prior). Consequently, the distribution of the statistic across the replica ensemble is the Bayesian posterior of the statistic.

Appropriate multinomial weights are generated by randomly sampling the sequences, with replacement. The Dirichlet random variants are generated by sampling N intervals between $N-1$ sorted random numbers uniformly distributed on the interval $[0,N]$ (Rubin, 1981). The requisite pseudo-random numbers were drawn from the Mersenne Twister generator (Matsumoto and Nishimura, 1998)

In the unweighted case, each correctly deduced sequence relationship contributes one divided by the number of possible sequence relationships to the coverage. The total possible number of correct sequence relationships is n^2-n, n being the number of sequences in the database, so the coverage contributed by one correctly deduced sequence relationship results in $(n^2-n)^{-1}$ coverage. The linear and quadratic normalization schemes weight the value of a correctly deduced sequence relationship by $s-1$ and s^2-s, respectively, where s is the number of sequences in the query sequence's superfamily. Consequently, the effective number of possible sequence relationships are also reduced. The coverage contribution for a correctly deduced sequence relationship is described by the following formulas:

	Standardbootstrap	Bayesianbootstrap
No normalization	$\frac{1}{n^2-n}$	$\frac{w_i w_j}{\left(\sum_{k=1}^{n} w_k\right)^2 - \sum_{k=1}^{n} (w_k)^2}$
Linear normalization	$\frac{1}{n(s-1)}$	$\frac{w_i w_j}{\left(\sum_{k=1}^{s}\right)\left(\left(\sum_{k=1}^{s} w_k\right) - w_i\right)}$
Quadratic normalization	$\frac{1}{(s^2-s)S}$	$\frac{w_i w_j}{\left(\left(\sum_{k=1}^{s}\sum_{k=1}^{s} w_k \cdot w_l\right) - \sum_{k=1}^{n} (w_k)^2\right) S}$,

where w_i is the weight of the query sequence, w_j the weight of the target sequence and S the number of superfamilies in the database. Summing to s indicates that only the weights of the sequences in the query sequence's superfamily should be summed. It should be clear that in the special case of

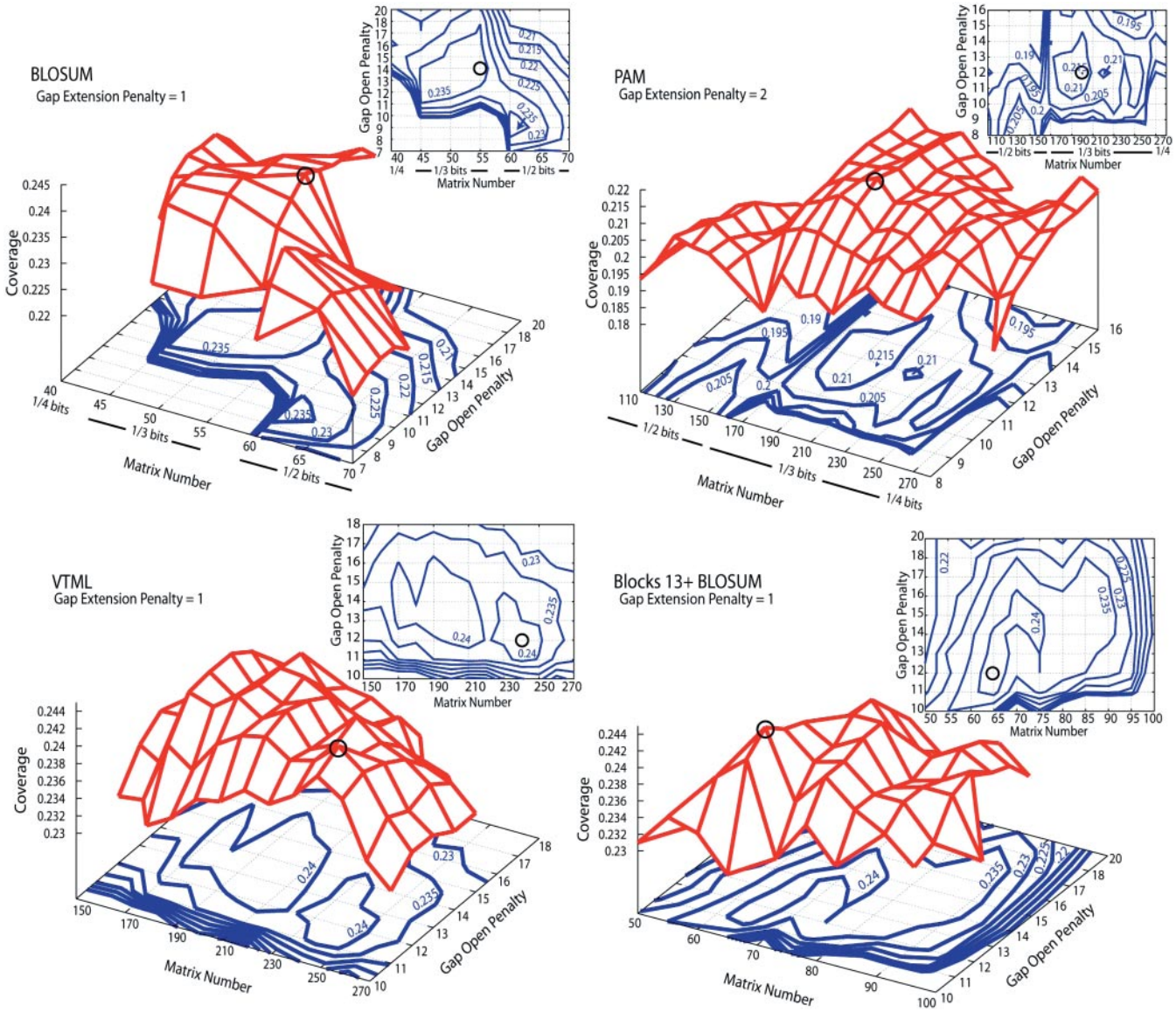

Fig. 3. Plots of the coverage versus matrix number and gap open penalty for each matrix family. The global maxima are indicated by circles and detailed in Table 1. In general, performances are robust to small changes away from the optimal parameters. All results are at 0.01 EPQ, use the optimal gap extension parameter setting and are under linear normalization on the training database. The main plots show the results in three dimensions, with a contour plot projected. The contour plot is also shown above each figure for clarity. High numbered PAM and VTML matrices represent large evolutionary times, whereas high numbered BLOSUM matrices represent short evolutionary times. For the standard, publicly available BLOSUM and PAM matrices, the matrix scaling varies with matrix number, as indicated, which results in discontinuities in the coverage surfaces. Small gap parameters represent more gappy alignments, and therefore, more distantly related sequences.

unitary weights, the Bayesian bootstrap formulas reduce to those of the standard bootstrap.

As previously discussed, and illustrated in Fig. 2, although the non-parametric and Bayesian bootstraps are formally equivalent in the large dataset limit, for our particular application and dataset, the non-parametric bootstrap produces a very biased replica ensemble. Consequently, estimates of statistical significances were carried out using the Bayesian bootstrap. For each bootstrap replica we calculate the difference in coverage between each search method, typically at 0.01 EPQ with linear normalization. From the set of coverage differences, we calculate the *Z*-statistic: the mean divided by the standard deviation. Generating 500 bootstrap replicas were found to be adequate.

RESULTS

For each of the four matrix families (PAM, BLOSUM from BLOCKS 5 and 13+, and VTML), we evaluated the coverage produced by SSEARCH, a standard implementation of the Smith–Waterman alignment algorithm with statistical scores (Pearson, 1991), on the training dataset under linear normalization at 0.01 EPQ. We generated BLOCKS 13+ BLOSUM and VTML matrices with software supplied by Henikoff (http://blocks.fhcrc.org, Blimps v3.5) and Muller (Muller *et al*., 2002), respectively, using 1/3 bit scaling consistently. We obtained BLOSUM matrices

Table 1. Optimal matrix and gap parameters for each matrix family and corresponding performance on the training and test databases under linear normalization at 0.01 EPQ

Matrix family	Matrix number	Gap open	Gap extension	Training dataset coverage	Test dataset coverage
BLOCKS 13+ BLOSUM	65	12	1	24.4	25.6 ± 0.09
VTML	240	12	1	24.3	25.2 ± 0.09
BLOSUM	55	14	1	23.9	25.2 ± 0.09
PAM	200	12	2	21.5	23.5 ± 0.09

Standard deviations derived from bootstrapping the test datasets are given.

from the internet distribution also at http://blocks.fhcrc.org and PAM matrix generation is described in Dayhoff (1978). The latter two matrix sets change scale with the matrix number. We varied the values of three parameters: matrix number, gap open penalty and gap extension penalty. Matrix number ranged from 40 to 100 for the BLOCKS 13+ BLOSUM matrices, 50 to 350 for the VTML matrices, 30 to 100 for the BLOSUM matrices and 10 to 310 for the PAM matrices. The gap open and extension penalties ranged, respectively, from 5 to 20 and 1 to 3. In total, we generated over 1200 result sets for the four different families at various different matrix and gap parameters. The performance of each matrix family was judged by the coverage at 0.01 EPQ on a test dataset, using the parameters that optimize the coverage on the training set. The best scoring matrix, with coverage of 25.6%, was BLOSUM65 derived from BLOCKS 13+ with gap open/extension parameters of 12/1. The VTML240/12/1 and standard BLOSUM55/14/1 matrices scored equivalently at 25.2%, and the PAM200/12/2 matrix scored at 23.5% (Table 1). In other words, the BLOCKS13+ BLOSUM matrices outperformed the VTML and standard BLOSUM matrices by 1.6%, and the PAM matrices by 8.5%. Considering the maturity of this particular technology, these gains are significant.

Figure 3 illustrates the variation of performance (as measured by linear coverage at 0.01 EPQ) with matrix and gap parameters. The contours mark a difference of 0.5% in coverage, which is roughly the minimum difference in mean coverage between statistically different methods and parameters (see Discussion section). In general, the optimum performances are robust to small changes in parameters. For example, the coverage difference between gap opening penalty settings in the range from 11 to 15 is only 0.5% when testing the VTML matrix family with the gap extension penalty set to 1 and the matrix number set to 140. It is also interesting to note that a relatively small reduction in gap parameters can lead to a large drop in homology detection coverage, presumably because alignment becomes overly permissive, allowing many gaps. For example, dropping the gap parameter from 11 to 10 in the previous example reduces coverage by 1.5%.

The top graphs in Figure 2 show the CVE plots for the optimal BLOCKS 13+ BLOSUM65 matrix. The bold CVE line is generated from the original data, whereas each of the lighter lines are generated from a bootstrap replica; a standard non-parametric bootstrap is in panel A and the Bayesian bootstrap is in panel B. The lower graphs show the coverage distribution of the bootstrap replicas at 0.01 EPQ. As can be seen, both the non-parametric and Bayesian bootstraps generate approximately Gaussian distributions of the replicates' coverage. However, the Efron bootstrap replicas are clearly biased relative to the original data. This is particularly notable for linear and quadratic normalizations, which emphasize the contributions of smaller superfamilies. In contrast, the Bayesian bootstrap ensemble does not exhibit a significant bias. Also notable is that the Bayesian coverage distributions are narrower. Both the bias and broadening effects can be seen to arise from the interaction of standard bootstrap resampling and small superfamilies. As we have discussed, each standard replica will randomly drop, on average, over one half of all size-2 superfamilies (and smaller, but still significant portions of other small superfamilies). This noisy sampling leads to the observed bias and larger variance of coverage, particularly when alternative normalizations emphasize small superfamilies. Consequently, in the following differential analysis of substitution matrix performance, we only consider the Bayesian bootstrap results.

Figure 4A displays coverage versus errors for all four families of matrices with optimal parameters under linear normalization. Clearly, the widths of these distributions are large compared with the average differences in coverage. This is emphasized in Figure 4B, which displays the bootstrap replicate distribution for each family at 0.01 EPQ. These overlapping distributions make it difficult to distinguish the performance of BLOSUM, VTML and BLOCKS 13+ BLOSUM matrices, although PAM is clearly worse than the other three. However, the statistic of interest in this analysis is not the *difference in mean coverage*, rather it is the *mean difference in coverage*, as shown in Fig. 4C. This distinction is significant, since the results obtained from a single data replica are correlated across different parameters. In our previous work (Green *et al*., 2002), we did not take this issue into account and generated independent bootstrap replicates for different methods, which resulted in an unnecessary reduction in sensitivity and an underestimation of statistical significance.

The absolute value of the *Z*-statistic (mean divided by the standard deviation of the difference in coverage) for each pair of search methods is shown in Table 2 as produced by the bootstrap. If the *Z*-statistic is >1.96, we reject the hypothesis that the methods possess equivalent performance at 95% confidence. It is clear from Table 2 that examining the appropriate metric, mean coverage difference, rather than the difference in mean coverage, yields a test that is more sensitive by a factor of >3 in the *Z*-score. To summarize the results, the least effective family is clearly PAM; BLOSUM and VTML are statistically indistinguishable; and the updated BLOCKS 13+ BLOSUM is significantly better than standard BLOSUM, but not quite significantly improved over VTML.

DISCUSSION

We have compared the performance of four substitution matrix families—PAM, BLOSUM, BLOCKS 13+ BLOSUM and VTML. Using the SSEARCH sequence comparison program, we evaluated each sequence in a database against every other sequence in that database. The Dayhoff PAM matrices are clearly worse than any other family, but this is well known and not surprising since PAM was trained on a small collection of relatively close homologs. The VTML family, which is essentially a modern reparameterized PAM, performs significantly better. However, surprisingly, it does not outperform the empirical BLOSUM families. This suggests that

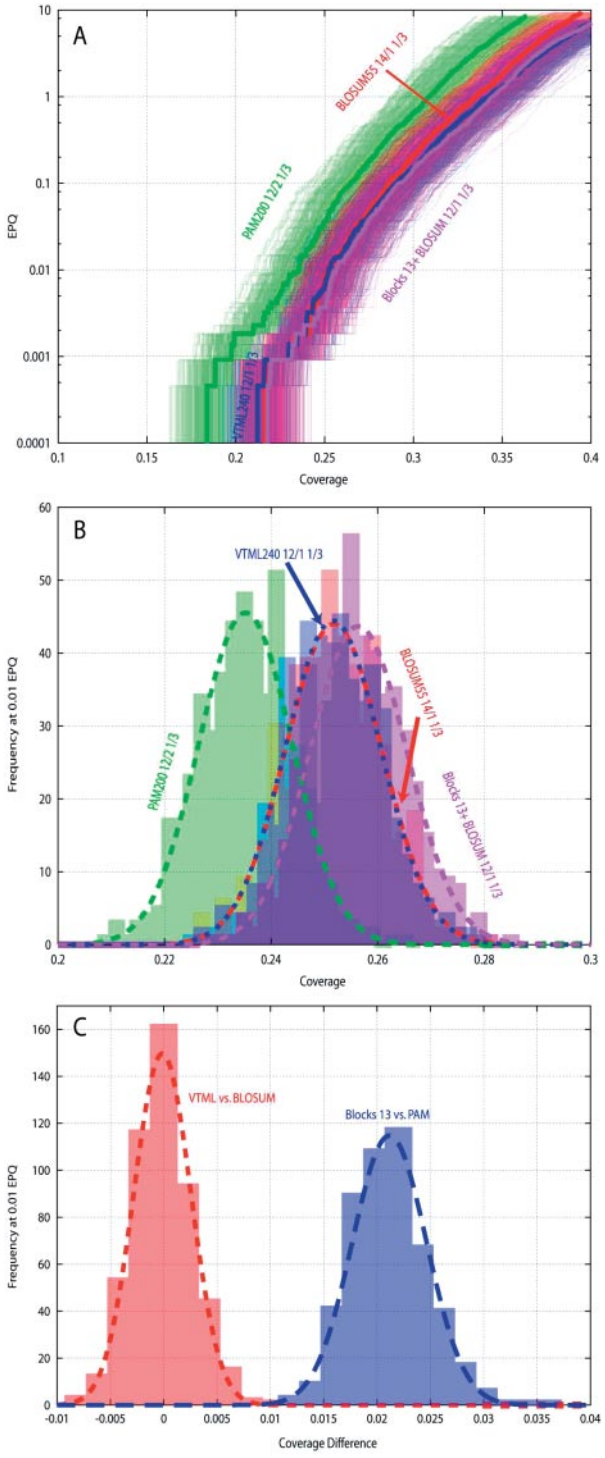

Fig. 4. Results of the Bayesian bootstrap applied to the optimal search methods. (**A**) The original CVE lines and bootstrapped lines. The original lines are heavier and thicker than the bootstrapped replicas. (**B**) The bootstrap distributions for each method at 0.01 EPQ and under linear normalization. Note that all the distributions overlap to some extent, and all but PAM overlap heavily. (**C**) Distributions of the bootstrap replicas' differences for the two extreme cases: VTML versus BLOSUM, which are nearly equivalent, and BLOCKS 13+ BLOSUM versus PAM, which differ significantly. The other four distributions are omitted for clarity.

Table 2. Absolute value of the *Z*-statistic for each pair of distributions of the optimal search methods at 0.01 EPQ and under linear normalization

Matrix set	Matrix number	BLOCKS 13+ BLOSUM	BLOSUM	VTML	PAM
BLOCKS 13+ BLOSUM	65	—	0.36/**2.16**	0.37/**1.89**	1.67/**6.09**
BLOSUM	55		—	0.01/**0.05**	1.31/**4.70**
VTML	240			—	1.31/**4.70**
PAM	200				—

Statistics are for difference in mean coverage (independent bootstrap replica)/mean difference in coverage. The latter is the statistic of interest.

the improvement of VTML and BLOSUM over PAM is the result of training on larger sets of remote homologs, rather than the rigor or sophistication of the training procedure. Clearly, there is a benefit to using the more recently developed matrix sets rather than the standard BLOSUM and PAM matrices.

We have also demonstrated that the Bayesian bootstrap can be used to estimate statistical errors and intervals in a database homology search without the anomalies introduced into this same analysis by the standard bootstrap. This is because the Bayesian resampling does not underrepresent small superfamilies in the resampled replicas, and is therefore not subject to the bias and noise introduced by Efron's non-parametric bootstrap. It has been shown that the Bayesian and Efron's bootstraps are asymptotically equivalent for large datasets (Lo, 1987) and thus the Bayesian and standard bootstraps can be interchanged in principle. However, this result clearly does not hold in our case. This is because, although our datasets are relatively large, the prevalence of small superfamilies introduces a fine-grained structure to our data that precludes the application of the asymptotic limit. Since several researchers predict that most or all of the superfamilies remaining to be discovered contain comparatively few sequences (Brenner *et al*., 1997, 1998; Zhang and Delisi, 1998; Govindarajan *et al*., 1999; Coulson and Moult, 2002; Koonin *et al*., 2002), the coverage bias inherent to Efron's bootstrap will probably be exacerbated, rather than diminished, as sequence databases grow. The Bayesian bootstrap does not explicitly take the superfamily structure into account, but it is nonetheless robust to the superfamily sampling problem. Generally, the Bayesian bootstrap may exhibit similar advantages over the standard bootstrap whenever the underlying data has a fine-grained structure and the statistic of interest is sensitive to that structure. Moreover, with the Bayesian bootstrap we also gain an unambiguous Bayesian interpretation of resampling without increasing the computational or conceptual complexity.

ACKNOWLEDGEMENTS

We thank Sandrine Dudoit and Mark van der Laan for their valuable discussions regarding the statistical portions of this paper. Support for this work was provided by NIH grants 1 K22 HG00056 and 5 T32 HG00047, the Searle Scholar's Program (01-L-116) and the Sloan Postdoctoral Fellowship in Computational Molecular Biology. Funding to pay the Open Access publication charges for this article was provided by NIH grant K22 HG00056.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. (1993) A protein alignment scoring system sensitive at all evolutionary distances. *J. Mol. Evol.*, **36**, 290–300.

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Blake,J.D. and Cohen,F.E. (2001) Pairwise sequence alignment below the twilight zone. *J. Mol. Biol.*, **307**, 721–735.

Brenner,S.E. (1996) Molecular propinquity: Evolutionary and structural relationships of proteins. Cambridge University.

Brenner,S.E. *et al.* (1997) Population statistics of protein structures: lessons from structural classifications. *Curr. Opin. Struct. Biol.*, **7**, 369–376.

Brenner,S.E. *et al.* (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl Acad. Sci. USA*, **95**, 6073–6078.

Brenner,S.E. *et al.* (1995) Gene duplications in *H.influenzae*. *Nature*, **378**, 140.

Brenner,S.E. *et al.* (2000) The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res.*, **28**, 254–256.

Chandonia,J.M. *et al.* (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res*, **32**, D189–D192.

Chandonia,J.M. *et al.* (2002) ASTRAL compendium enhancements. *Nucleic Acids Res.*, **30**, 260–263.

Coulson,A.F.W. and Moult,J. (2002) A unifold, mesofold and superfold model of protein fold use. *Proteins*, **46**, 61–71.

Crooks,G.E. and Brenner,S.E. (2004) An alternative model of amino acid replacement. *Bioinformatics*, **21**, 975–980.

Crooks,G.E., Green,R.E. and Brenner,S.E. (2005) Pairwise alignment incorporating dipeptide covariation. *Bioinformatics*, In Press.

Dayhoff,M.O. (1978) A Model of Evolutionary Change in Proteins. *Protein Sequence and Structure Vol 5, Suppl*, **3**, 345–352.

Durbin,R., Eddy,S.R., Krogh,A. and Mitchison,G.J. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK.

Efron,B. (1979) 1977 Rietz Lecture—Bootstrap Methods—Another Look at the Jackknife. *Ann. Stat.*, **7**, 1–26.

Efron,B. and Robert,J.T. (1993) *An Introduction to the Bootstrap*. Chapman & Hall/CRC, Boca Raton.

Geetha,V. *et al.* (1999) Comparing protein sequence-based and predicted secondary structure-based methods for identification of remote homologs. *Protein Eng.*, **12**, 527–534.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Govindarajan,S. *et al.* (1999) Estimating the total number of protein folds. *Proteins*, **35**, 408–414.

Green,R.E. and Brenner,S.E. (2002) Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison. *Proc. IEEE*, **90**, 1834–1847.

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.

Henikoff,J.G. *et al.* (2000) Increased coverage of protein families with the blocks database servers. *Nucleic Acids Res.*, **28**, 228–230.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Henikoff,S. and Henikoff,J.G. (1993) Performance evaluation of amino acid substitution matrices. *Proteins*, **17**, 49–61.

Karplus,K. *et al.* (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.

Koonin,E.V. *et al.* (2002) The structure of the protein universe and genome evolution. *Nature*, **420**, 218–223.

Lindahl,E. and Elofsson,A. (2000) Identification of related proteins on family, superfamily and fold level. *J. Mol. Biol.*, **295**, 613–625.

Lo,A.Y. (1987) A large sample study of the Bayesian bootstrap. *Ann. Stat.*, **15**, 360–375.

Matsumoto,M. and Nishimura,T. (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Trans. Model. Comput. Simul.*, **8**, 3–30.

Muller,T. *et al.* (2002) Estimating amino acid substitution models: a comparison of Dayhoff's estimator, the resolvent approach and a maximum likelihood method. *Mol. Biol. Evol.*, **19**, 8–13.

Muller,T. and Vingron,M. (2000) Modeling amino acid replacement. *J. Comput. Biol.*, **7**, 761–776.

Murzin,A.G. *et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Park,J. *et al.* (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.*, **284**, 1201–1210.

Pearson,W.R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, **11**, 635–650.

Pearson,W.R. (1995) Comparison of methods for searching protein sequence databases. *Protein Sci.*, **4**, 1145–1160.

Pearson,W.R. (1996) Effective protein sequence comparison. *Methods Enzymol.*, **266**, pp. 227–258.

Pearson,W.R. (1998) Empirical statistical estimates for sequence similarity searches. *J. Mol. Biol.*, **276**, 71–84.

Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Rubin,D.B. (1981) The Bayesian bootstrap. *Ann. Stat.*, **9**, 130–134.

Russell,R.B. *et al.* (1997) Recognition of analogous and homologous protein folds: analysis of sequence and structure conservation. *J. Mol. Biol.*, **269**, 423–439.

Schaffer,A.A. *et al.* (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.

Schaffer,A.A. *et al.* (1999) IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics*, **15**, 1000–1011.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Zachariah,M.A. *et al.* (2004) A generalized affine gap model significantly improves protein sequence alignment accuracy. *Proteins*, **58**, 329–338.

Zhang,C. and DeLisi,C. (1998) Estimating the number of protein folds. *J. Mol. Biol.*, **284**, 1301–1305.